In the Claims

1. (Previously Presented) A method for graphically representing object oriented programming logic, the method comprising the steps of:

(1) providing a plurality of different symbols for use in a diagram of object oriented programming logic, each different symbol representing a different type of object in object oriented programming;

(2) selecting an object as a main object of the logic to be represented in the diagram;

(3) drawing a symbol corresponding to the main object and labeling the symbol with a label descriptive of the object's features so that it is distinguishable from other symbols of the same object type;

(4) for each object assigned to or defined within the main object, drawing a symbol corresponding to that object and labeling the symbol with a label descriptive of the object's features; and

(5) drawing a line between each object drawn in step (4) and another object in the graphical representation to which it is assigned or within which it is defined.

2. (Previously Presented) The method of claim 1 further comprising the step of:

(6) providing a plurality of additional different symbols for use in the diagram, each of the additional different symbols representing a different an object oriented programming element other than an object.

3. (Original) The method of claim 1 further comprising the step of:

(7) graphically denoting the main object in the diagram by drawing another symbol around the symbol for the main object.

4. (Previously Presented) The method of claim 3 wherein step (7) comprises drawing a circle completely enclosing the symbol of the main object.

5. (Original) The method of claim 1 wherein the labels comprise text.

6. (Original) The method of claim 1 wherein step (5) comprises drawing the line between the object defined in step (4) and another object it is most directly assigned to or is most directly defined within.

7. (Original) The method of claim 1 wherein the method is used to document software.

8. (Original) The method of claim 1 wherein the method is used to prepare a program specification.

9. (Original) The method of claim 1 further comprising the step of:

(8) repeating steps (1) – (5) to prepare a plurality of separate diagrams corresponding to separate parts of an overall application and wherein a first

object is the main object appearing in at least a first one of the diagrams and is not a main object appearing in at least a second one of the diagrams.

10. (Original) The method of claim 9 wherein the second diagram does not disclose objects assigned to and defined within the first object and the first diagram does disclose objects assigned to and defined within the first object.

11. (Original) The method of claim 10 wherein the second diagram is an application-level representation disclosing an overall software system.

12. (Original) The method of claim 10 wherein the label for the first object in the second diagram identifies the first diagram as disclosing further details of the first object.

13. (Original) The method of claim 1 wherein the symbols representing different object
types include:

a first symbol for representing objects that are application type objects;

a second symbol for representing objects that are window type objects;

a third symbol for representing objects that are class type objects;

a fourth symbol for representing objects that are event script type objects; and

a fifth symbol for representing objects that are method type objects.

14. (Previously Presented) The method of claim 2 wherein the symbols representing different object types include:

a first symbol for representing objects that are application type objects;

a second symbol for representing objects that are window type objects;

a third symbol for representing objects that are class type objects;

a fourth symbol for representing objects that are event script type objects; and

a fifth symbol for representing objects that are method type objects.

and wherein the symbols representing additional program elements include:

a sixth symbol for representing data transfer;

a seventh symbol for representing databases;

an eighth symbol for representing remote links; and

a ninth symbol for representing inheritance.

15. (Original) The method of claim 14 wherein the sixth, eighth, and ninth symbols are drawn connecting two other object symbols.

16. (Original) The method of claim 14 wherein the symbols representing different object types further include;

a tenth symbol for representing objects that are menu type objects;

a eleventh symbol for representing objects that are frame type objects;

an twelfth symbol for representing objects that are button type objects;

a thirteenth symbol for representing objects that are data structure type objects; and

a fourteenth symbol for representing objects that are not one of the other object types.

17. (Original) The method of claim 13 further comprising the step of:

(9) providing in a separate document a description of the logic to be performed responsive to an event script.

18. (Original) The method of claim 13 wherein the fourth symbol representing event

script type objects is drawn connected to another object that directly executes the event script corresponding to the event script symbol.

19. (Original) The method of claim 13 wherein the fifth symbol representing method type
objects is drawn connected to the main object of the diagram and represents that the object is available within that main object and does not represent that the main object invokes it.

20. (Original) The method of claim 1 wherein the method is implemented via a computer program, and wherein step (1) comprises providing a graphical user interface in which a user is presented with a pallet containing the symbols and wherein steps (3) and (4) comprise dragging and dropping the symbols from the pallet into a work area.

21. (Original) The method of claim 1 wherein the method is implemented via a computer program, and wherein step (1) comprises providing a graphical user interface in which a user is presented with a pallet containing the symbols and wherein steps (3) and (4) comprise dragging and dropping the symbols from the pallet into a work area, and wherein the labels comprise text and further wherein at least some of the text labels are hidden text that can be made to appear in the graphical representation via an action taken by a user.

22. (Currently Amended) A computer readable product embodied on computer readable media readable by a computing device for enabling a user to

generate a graphical representation of object oriented programming logic, the product comprising:

first computer executable instructions that provide a graphical user interface in which a user is presented with a plurality of different symbols for use in developing a graphical representation of object oriented programming logic, each different symbol representing a different type of object in object oriented programming;

second computer executable instructions that enable the user to drag and drop symbols into a workspace and label the symbol<u>s</u> with a label descriptive of the object's features; and

third computer executable instructions that enable the user to draw lines between objects in the workspace.

23. (Original) The computer readable product of claim 22 further comprising:

fourth computer executable instructions that enable the user to graphically denote the main object in the diagram by drawing another symbol around the symbol for the main object.

24. (Original) The computer readable product of claim 22 further comprising:

fifth computer executable instructions that enable the user to denote one and only one object in the workspace as a main object.

25. (Original) The computer readable product of claim 24 wherein the fourth computer executable instructions comprise instructions enabling the user to enclose the one and only one object within a circle.

26. (Original) The computer readable product of claim 22 wherein the labels are text labels.

27. (Original) The computer readable product of claim 22 further comprising:

sixth computer readable instructions that enable the user to prepare a plurality of the diagrams corresponding to separate parts of an overall application and further comprising computer readable instructions for enabling the user to specify relationships between individual ones of the diagrams.

28. (Original) The computer readable product of claim 27 wherein the sixth computer readable instructions comprise instructions that enable the user to including references associated with symbols in one diagram identifying at least one other diagram within which the object represented by that symbol also appears.

29. (Original) The computer readable product of claim 28 wherein the sixth computer readable instructions comprise instructions that enable the user to

specify in a first one of the diagrams the nature of the relationship of the representation of the object in the first diagram relative to the representation of the object in a second diagram, wherein the relationship between the object as represented in the first and second diagrams is selected from the group comprising: (1) the second diagram discloses additional details about the object in the first diagram; (2) the second diagram shows the object in a more abstract context than the first diagram; and (3) the object is the main object of the second diagram.

30. (Original) The computer readable product of claim 22 wherein the symbols representing different object types include:

a first symbol for representing objects that are application type objects;

a second symbol for representing objects that are window type objects;

a third symbol for representing objects that are class type objects;

a fourth symbol for representing objects that are event script type objects;

and

a fifth symbol for representing objects that are method type objects.

31. (Original) The computer readable product of claim 24 wherein the symbols representing different object types include:

a first symbol for representing objects that are application type objects;

a second symbol for representing objects that are window type objects;

a third symbol for representing objects that are class type objects;

a fourth symbol for representing objects that are event script type objects; and

a fifth symbol for representing objects that are method type objects.

and wherein the additional symbols representing additional program elements include:

a sixth symbol for representing data transfers;

a seventh symbol for representing databases;

an eighth symbol for representing remote links; and

a ninth symbol for representing inheritance.

32. (Original) The computer readable product of claim 31 further comprising:

seventh computer executable instructions that restrict the user to using the sixth, eighth, and ninth symbols to connect two other object symbols.

33. (Original) The computer readable product of claim 30 wherein the symbols representing different object types further include;

a tenth symbol for representing objects that are menu type objects;

an eleventh symbol for representing objects that are frame type objects;

a twelfth symbol for representing objects that are button type objects;

a thirteenth symbol for representing objects that are data structure type objects; and

a fourteenth symbol for representing objects that are not one of the other object types.

34. (Original) The computer readable product of claim 30 further comprising:

eighth computer executable instructions that enable the user to providing in a separate document a description of the logic to be performed responsive to an event script.

35. (Original) The computer readable product of claim 22 further comprising:

ninth computer executable instructions that enable the user to insert hidden text associated with symbols in the workspace that can be made to appear in the workspace responsive to an action taken by a user.